Analyzing Web Profiles using Probabilistic Ontologies

Pawel Kozak
IRCLOVE Community Portal
Heinrich-Heine-Str. 8
63071 Offenbach am Main
Germany
kozak@irclove.de

Karsten Tolle
Databases and Information Systems (DBIS)
Johann Wolfgang Goethe-University
60325 Frankfurt am Main
Germany
tolle@dbis.informatik.uni-frankfurt.de

ABSTRACT
In this paper, we discuss our probabilistic ontological solution for analysis of Web profiles. The analysis of Web profiles is a very demanding and multi-layered task; especially probabilistic information in terms of probability distributions and weights is often the key to an expressive analysis. In our research we designed the Probabilistic Profile Analysis Ontology (*PPAO*) using Markov Logic Networks (*MLNs*) afterwards we conducted experimentation to evaluate the scalability and expressiveness of this solution. MLNs were chosen as the underlying formalism because of their probabilistic nature, intuitive and expressive modeling ability due to first order logic as well as ability to use approximation algorithms to improve reasoning performance. A similarity benchmark between profiles was designed within the PPAO concept as a special task and real static and probabilistic data from the German IRCLOVE© community www.irclove.de was used for the analysis.

Although our solution turned out to achieve expressive results, the experimentation revealed a mixed picture on scalability of profile analyzing with MLNs. Therefore we will discuss these results and propose possible research directions for further improvements on using probabilistic ontologies for profile analysis.

Categories and Subject Descriptors
I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving – inference engines, uncertainty, "fuzzy" and probabilistic reasoning; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – Predicate Logic, Representation Languages, Semantic Networks; H.3.4 [**Information Storage and Retrieval**]: Systems and Software – User profiles and alert services, Performance evaluation (efficiency and effectiveness)

General Terms
Measurement, Performance, Experimentation, Design.

WebSci '11, June 14-17, 2011, Koblenz, Germany.

Keywords
Probabilistic Ontologies, Markov Logic Networks, Trust, User Profiles.

1. INTRODUCTION
Imagine, in a large and complex Web community environment you want to propose users to get in contact with others that are potentially interesting for them – how effectively can it be achieved by using ontologies and related techniques?

In today's Web, profiles of users represent the core of all community-driven personalized applications – this range from social networks like Facebook™ to major Web enterprises like Ebay™ or Amazon™. These profiles describe not only personal data but also user behavior, mostly provided by interaction with the specific application, e.g. persons can purchase items, read articles, write comments or communicate with each other. Therefore, analyzing of the interconnections between profiles is an important task in order to extract valuable data which will become even more important with ongoing development of sophisticated semantic enhancements of the Web.

In many cases there are multiple resources providing data for one application, as for instance seen in Facebook Apps. However, in order not to harm data quality there is a significant demand for trust regulation when enriching profile information. In particular, data from the less trustworthy sources should be eliminated or its effects should be limited by weight-based mechanisms.

With these demanding tasks in mind we designed and implemented the Probabilistic Profile Analysis Ontology (*PPAO*) based on Markov Logic Networks (MLNs) [2]. The abstract PPAO concept insists of three core elements: *static* and *dynamic references*, which are classes of rarely resp. continuously changing reference points (for instance address data or article categories), and *users*. The core elements are connected with each other by binary or probabilistic roles that represent a possible interaction between the elements. For example "view" would be a probabilistic role between the class of users and the dynamic reference "article category". Full implementation of PPAO including the inference-specific part is described in [4]. In this paper we will present and discuss some of our evaluations on analyzing real-world profiles using PPAO. Figure 1 shows the full workflow from a community-like domain to the final task solution. We will also briefly discuss in section two why we choose MLNs and why the results are fitting better our use case needs compared to existing distance metrics. Section 3 provides details of our experiments on real data. Finally, section 4 closes this paper with our conclusions.

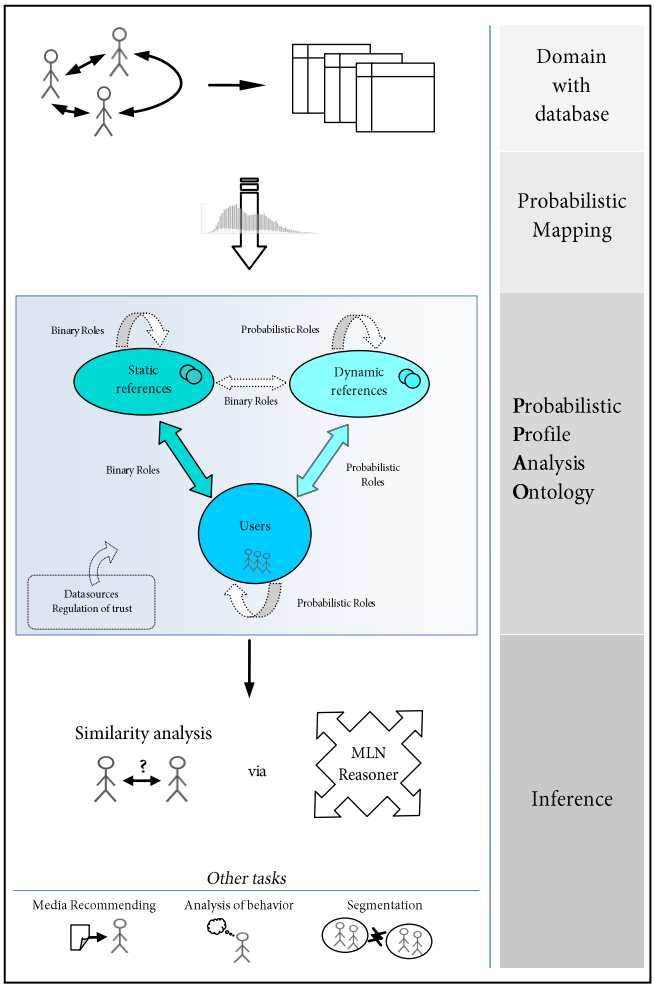

Figure 1. Workflow of our solution for analyzing profiles and its PPAO main component implemented with MLNs.

2. USE CASE AND IMPLICATIONS

Our evaluations rely on real data from the German IRCLOVE© community irclove.de. In IRCLOVE a user has the ability to create a multimedia library and execute different actions like creating, reading or commenting media. With over 32.000 users and various categories the complexity increases dramatically. Common knowledge representation formalisms like description logics (DLs) or the Web Ontology Language (OWL) suffer under the bad time complexity [7] and an optimum between expressiveness and tractability of those formalisms is still an unresolved research problem [1]. Therefore, reasoning over such huge amounts of data with classic DL-based formalisms is definitely intractable. Those formalisms were designed to handle static information. But this is not sufficient to describe situations in complex domains where user behavior within the system can be interpreted as probability distribution, for instance reading articles in 20% and viewing photos in 80% of all cases.

Additionally, probabilistic mechanisms are required to weight information, which is a precondition for integration of trust regulation into ontologies.

In order to better deal with those probabilistic mechanisms, we investigated MLNs. They are development inside the statistic relational learning research field and combine the expressiveness of first-order logic with probabilistic elements and approximation algorithms. There exist some powerful MLN reasoners as PyMLN, Alchemy and TheBeast. They also provide approximating algorithms in order to cope with huge amount of data. Therefore, MLNs are one of very few practical applicable probabilistic formalisms, so far [5]. However, we will discuss the limits of our approach to use MLNs for profile matching analysis in section 3.

Another reason for us to use MLNs was due to the different result interpretation. When we thought of a recommender system for IRCLOVE, our goal was to recommend potential interesting users for someone – the so called master-user. In our view this is not someone whose profile is similar or even equal to the master-user.

Looking at the profiles of the users of our use case IRCLOVE, we realized that most users have a very diverse profile. In the profiles over 50 hobbies occurred, whereby the most frequent hobbies where music, sports and reading books. By using distance metric the highest match would be between users of similar profiles. Of course this would make sense; however, due to the diversity this is at least questionable. By using MLNs we aimed at finding for a certain profile those profiles that are more like opinion leaders or experts for the highest rated hobbies.

This means we are aiming to predict who will likely behave similar or has similar interests in future. Whereby, persons with special interests – a very high value for a certain category – should be ranked higher, if this category is accordingly relevant for the master-user.

This is exactly what MLNs do. As a small example think of three persons called Anna, Peter and Lena. They are requested to choose between two choices. From their past behavior we know that Anna and Peter behave extremely similar, let us say they pick to 80% the first choice and to 20% the second. Lena always picks the first choice. A recommender system would likely match Anna and Peter, because of the similar profiles. However, when you ask the question who will most likely pick the same choice in the next round, Lena has 80% correlation with Anna and Peter, while between Anna and Peter we only have 68% correlation.

Also in our experiments with real data this could be reached. One example is shown in Figure 2, where some of the top related profiles are shown regarding one master-user. As you can see the highest correlation is with those users that could be indicated as experts for those categories the master-user is most interested.

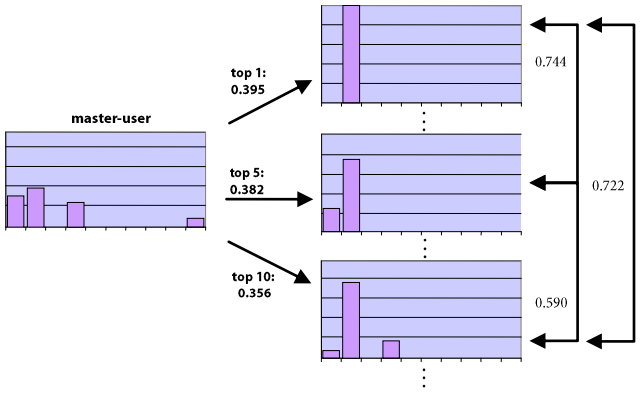

Figure 2. Visualized user profile (master-user) on the left, compared to those profiles that correlate best based on the MLN results.

3. EXPERIMENTATION

3.1 Methodology

In our experiments we examined the performance of the PPAO solution. As stated above there are some powerful MLN reasoners available, although they are still under development and are not bug-free. For our cause we decided to use PyMLN because of good usability, performance and stability with our MLN ontology. The used version number of PyMLN as a part of the ProbCog Software Suite [6] was 1573. Furthermore a Dual-Opteron Server with 4 GB RAM and a virtual Linux machine where used to run inference. The workflow was straightforward according Figure 1: PPAO was generated from the IRCLOVE database then imported into PyMLN afterwards the specific inference queries were executed. Although PyMLN offers MC-SAT and Gibbs Sampling as two approximating algorithms, only Gibbs Sampling worked for us whereas MC-SAT was stuck for a simple query. Thus for all experiments only Gibbs Sampling was used.

Two scenarios were set up for evaluation: in the ALL vs. ALL scenario the similarity index was calculated for a limited set of users and for each user pair. Hence the number of the calculated indexes for this scenario is $\binom{n}{2} = \frac{n \cdot (n-1)}{2}$ and for 1000 users one would have to calculate nearly a half million indexes. For IRCLOVE with 38.000 users an amount of over 72 millions indexes demonstrates the tremendous size of such problems. Therefore the softer 1 vs. ALL scenario was also evaluated, where one user was matched against a set of users resulting in n indexes.

3.2 Results

In the already mentioned IRCLOVE media library portal called "IRCLOVE Persönlichkeiten" each media is associated with a category and the categories are organized as taxonomy. Furthermore users can execute actions within the system. Since the similarity index has been designed in order to match users according to their areas of interest, statistic information from this portal was used to create PPAO. Especially actions were chosen as probabilistic roles and categories as dynamic references. Additionally hobbies written by the user into his profile were added as static references.

Figure 3 shows two major results of our experiments. The blue curve demonstrates the inference time of the 1 vs. ALL scenario with 1653 users, 4 actions and a set of 5 possible hobbies (see [4] for explanation why 5 is enough for effective matching) and increasing number of categories. Increased number of categories and actions as factors taken into account improves the expressiveness of the similarity index. So the good news is that the progression here is only linear. Furthermore the calculation remained linear during additional experiments in both other factors of precision, the number of actions and hobbies. The red curve is showing the inference time of the ALL vs. ALL scenario with one action and 10 categories. Obviously the calculation is polynomial in the number of users. Moreover we were not able to run the calculation of much more than 500 users because the process ran out of RAM at 3.5 GB on our test machine.

As for the RAM consumption we observed linear growth in the number of actions and users, ranging between 0.5 GB and 3.5 GB which mostly correlated with the number of ground atoms and formulas in PPAO ranging from 0.2 to 1.2 millions.

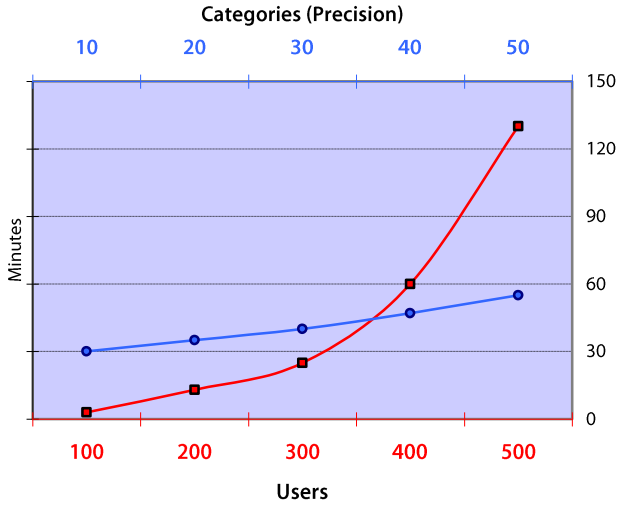

Figure 3: Blue curve reveals linear progression with increasing number of categories which improve precision. Red curve shows polynomial progression with increasing number of users in the ALL vs. ALL scenario.

4. CONCLUSIONS

In the introduction we raised the question: How to find within a community for one person potentially interesting persons he could contact – based on their behavior profile?

By using PPAO and MLNs we showed that we could find answers to this question that go beyond calculating distance metrics and are therefore more meaningful. Additionally with our experiments we conducted two major results: a) the scalability of PPAO was well in factors which improve expressiveness of the similarity index, b) we were only able to analyze a relatively small amount of about 500 users with our provided hardware. Superficially, the reason for this limitation is the huge size of the problem. Furthermore we observed that the time needed by PyMLN for reading and creating the MLN was at about 50% of total calculation time, although the size of the ontology stayed relatively small at 30 MB. This and other observations lead to the conclusion that for currently available MLN reasoners further optimization must be done for improving performance.

Definitely, our (already optimized) ontology allows to analyze some thousands profiles with appropriate expressiveness on better hardware. Nevertheless it is not possible to analyze medium and large sized communities. One way out would be to analyze only selected groups of profiles for instance only active users. A related approach that is already under research for exact MAP inference is using so called Top-K methods during the inference. This allows setting a boundary to sort out "bad" results, for instance profiles with low similarity [3].

However, another very important issue in our view is to address the implementation of parallelism into MLN reasoning, which is not supported by any reasoner so far. This would allow distributing the calculation on different machines and cores. This should enable one to scale the inference.

5. REFERENCES

[1] BAADER, F., HORROCKS, I., SATTLER, U.: *Description Logics*. In Staab, S., Studer, R. eds: Handbook on Ontologies. Berlin, Springer Verlag. (2009) 21-43

[2] DOMINGOS, P., RICHARDSON, M.: *Markov Logic: A Unifying Framework for Statistical Relational Learning*. In Getoor, L., Taskar, B. eds: Introduction to Statistical Relational Learning. USA, Cambridge, MIT Press. (2007) 339-373.

[3] NIEPERT, Mathias: *A Delayed Column Generation Strategy for Exact k-Bounded MAP Inference in Markov Logic Networks*. In proceedings of 26th UAI, USA, Catalina Island, AUAI Press. (2010)

[4] KOZAK, Pawel: *Ontologiebasierte Ontologiebasierte probabilistische Profilanalyse mit Markov Logic Networks*, Master‘s Thesis, Goethe University Frankfurt am Main. (2011) http://www.irclove.de/files/Kozak_Pawel_Thesis_2011.pdf

[5] PREDOIU, L., STUCKENSCHMIDT, H.: *Probabilistic Models for the Semantic Web – A Survey*. Technical Report, Mannheim, Universität Mannheim. (2008)

[6] *ProbCog: Probabilistic Cognition for Cognitive Technical Systems.* (2010) http://ias.cs.tum.edu/research-areas/knowledge-processing/probcog

[7] TOBIES, Stephan*: Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD Thesis, Aachen, RTWH Aachen. (2001)